



Technical Report

# Cloning SAP on Sybase Databases with NetApp Snap Creator

Steffen Dittmar, NetApp  
March 2012 | TR-4038

## Abstract

This document provides a method to automate application-consistent replication of a SAP<sup>®</sup> system on a Sybase database in a way that saves both time and storage space using NetApp<sup>®</sup> Snap Creator<sup>™</sup>.

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction</b> .....	<b>3</b>
1.1	Use Case: SAP Development and Test.....	3
1.2	Technology Used.....	4
<b>2</b>	<b>Solution Goals</b> .....	<b>4</b>
<b>3</b>	<b>Solution Details</b> .....	<b>5</b>
3.1	Overview.....	5
3.2	Environment .....	6
<b>4</b>	<b>Snap Creator Configuration</b> .....	<b>8</b>
4.1	Mount and Unmount Commands .....	8
4.2	Server Configuration Parameters .....	10
4.3	Agent Configuration .....	16
4.4	Running Snap Creator .....	16
<b>5</b>	<b>Conclusion</b> .....	<b>17</b>
	<b>Appendix</b> .....	<b>17</b>
	Sample Configuration .....	17
	Variables Used in this Document.....	20
	<b>References</b> .....	<b>20</b>

## LIST OF TABLES

Table 1)	FlexVol and qtree layout.....	6
Table 2)	Required options.....	10
Table 3)	Cloning options.....	12
Table 4)	NFS-specific options.....	12
Table 5)	Client/server configuration.....	13
Table 6)	Sybase settings. ....	13
Table 7)	Variable definitions. ....	20

## LIST OF FIGURES

Figure 1)	Traditional method of copying data between SAP systems: estimate 24 hours.....	3
Figure 2)	NetApp cloning: estimate 12 hours. ....	4
Figure 3)	Overview of the desired architecture.....	5
Figure 4)	SAP directories of the example source system.....	7
Figure 5)	Sybase directory after cloning. ....	10

# 1 Introduction

This report describes a method to replicate a production SAP system on a Sybase database to a development environment, both running on NetApp storage. This solution uses NetApp technology, such as NetApp Snap Creator and NetApp FlexClone®, to minimize the time and storage capacity required to refresh development and test environments.

This report describes the processes to:

- Set up an environment that allows complete database cloning using Snap Creator.
- Configure Snap Creator to create a clone of the entire database.
- Initiate cloning using Snap Creator.

## 1.1 Use Case: SAP Development and Test

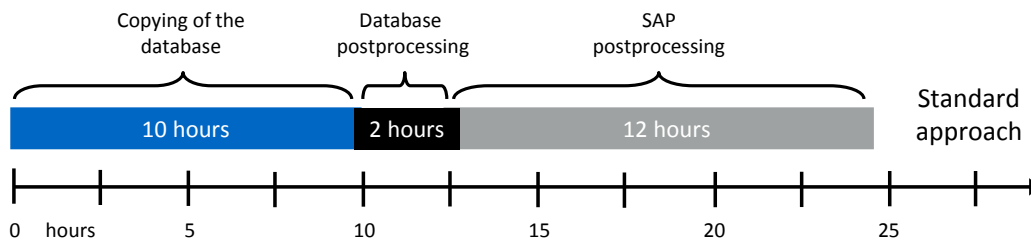
SAP software has a large customer base and plays a critical role in their businesses. To maintain the stability of systems running SAP software in the face of change, it is the SAP recommended best practice to implement a staged environment with multiple separate instances from development to test and from training and to production.

Providing the development, test, and training systems with data from the production instance poses the following challenges:

- **Storage capacity requirements.** Multiple copies of the data are required, which results in large storage capacity requirements.
- **Time to create development and test environments.** Making independent development and test copies available takes time, but it is crucial for incident management and acceleration of test procedures.
- **Effect on production systems.** The overhead of creating development and test environments must be very low so it does not affect production systems.

Figure 1 shows an example time exposure for updating a test system's data. The actual amount of time required depends on the size of the database and on the SAP postprocessing required. Postprocessing also depends on the customer's environment and therefore cannot be automated by Snap Creator.

Figure 1) Traditional method of copying data between SAP systems: estimate 24 hours.



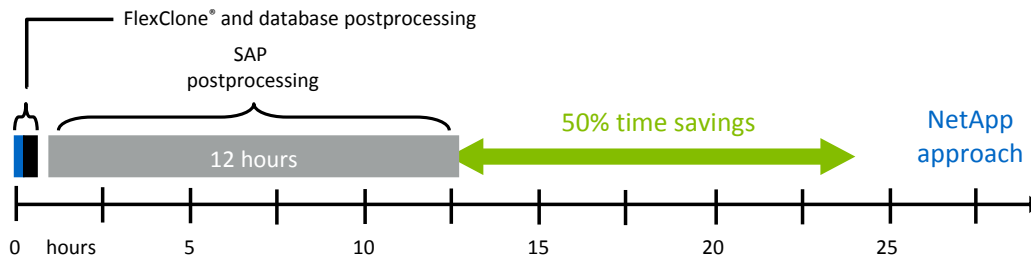
Using NetApp storage and features provides significant benefits in this scenario:

- **Storage savings.** With FlexClone, create copies of the production database using little to no additional storage capacity. Traditional methods require the same storage capacity as the production database for each additional development or test copy of the database.
- **Time savings.** Provision copies of the database almost immediately rather than in the hours or days required by traditional copying methods. This reduces time that database administrators require to provision test and development environments for developers and allows developers to have access to the up-to-date development environments much sooner.

- **Application consistency.** Snap Creator makes sure that the copied database is in a consistent state without taking the production system offline. The traditional approach of copying a running database introduces the risk of inconsistency, and taking the production system offline results in decreased production system availability.
- **Automation and quality.** The numerous steps required to integrate the copied database into a new environment are automatically performed by Snap Creator. Running these steps manually is time consuming and has the potential to introduce errors.
- **Unaffected production system.** FlexClone copies reference data instead of actually copying it. This relieves the production system of impairing input/output (I/O) accesses and high CPU use. Because no performance loss is expected on the production system, the test environment databases can be updated at any time. Traditional storage requires copies to be created only when no high-responsibility demands are posed on the production system, overnight, for example.
- **Multiple copies.** FlexClone provides more test environments that allow development and testing activities to run in parallel, which results in decreased time to market.

Figure 2 illustrates that although SAP postprocessing still takes the same amount of time, database copying and postprocessing happen almost instantly, which results in 50% time savings in this example.

Figure 2) NetApp cloning: estimate 12 hours.



## 1.2 Technology Used

This solution uses FlexClone and Snap Creator technologies.

- **Snap Creator Framework.** Snap Creator manages application and database consistency. It coordinates consistency actions and triggers the FlexClone procedure. The Sybase ASE plug-in of Snap Creator discussed in this document manages the proper handling of Sybase.
- **FlexClone.** FlexClone creates an instant, space-efficient, identical copy of selected source data on the storage controller.

## 2 Solution Goals

The solution described in this report provides a method for customers to replicate their production Sybase databases to development and test environments using NetApp storage.

The solution achieves the following goals:

- Clone a Sybase ASE database with minimal effort.
- Keep the database in a consistent state without taking it offline.
- Automatically integrate the cloned database into a new SAP system.

### 3 Solution Details

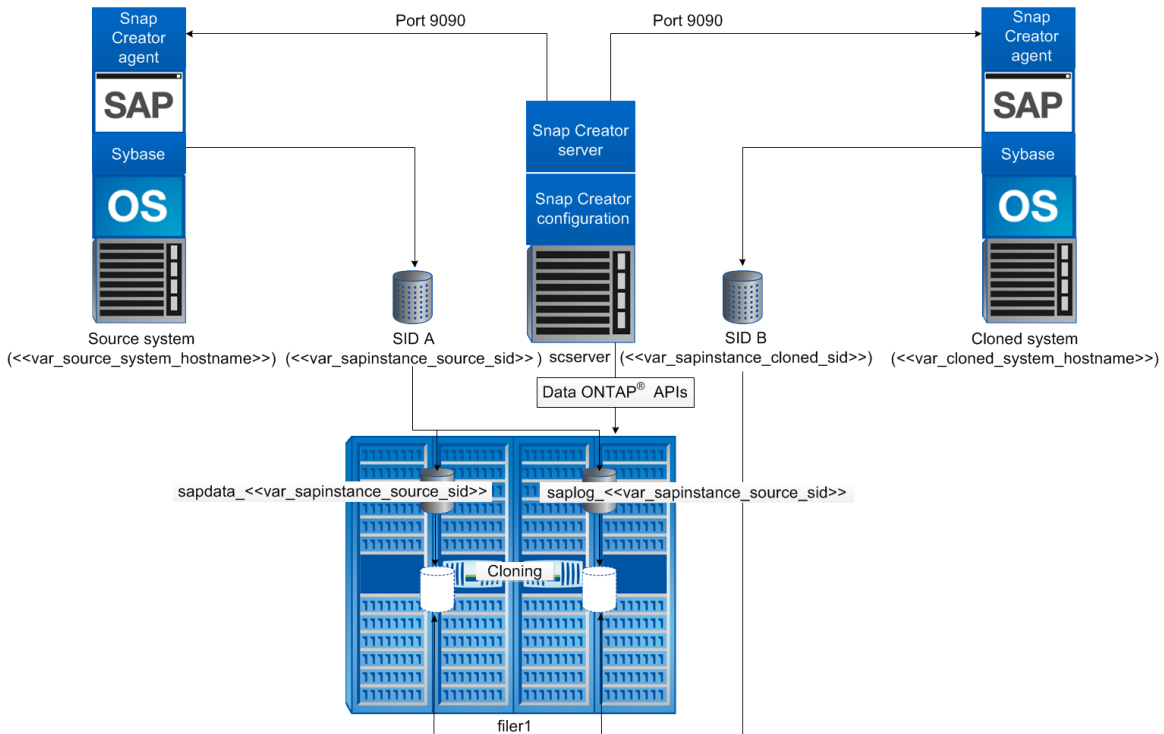
In this section, the assumed architecture and the underlying conditions that must be met are discussed in greater detail. This section provides an overview of the solution and discusses the requirements of implementing it.

#### 3.1 Overview

The following sections describe using Snap Creator and the Sybase plug-in to clone a Sybase ASE database from the source system on which it is running to another machine (the cloned system). This solution can also be used to easily create multiple clones for multiple cloned systems.

Figure 3 shows an overview of the components involved in the solution.

Figure 3) Overview of the desired architecture.



Snap Creator consists of a server and one or more agent instances that can, but do not have to, run on different machines. The server initiates all operations, and the agent is responsible for executing them, which makes it possible to store all configuration files on a single system, namely, the one running the server instance.

In the example architecture, there are a source system and a cloned system, each of which run SAP software. The source system is the production system, and the cloned system can be a test or development environment. Both machines are connected to a NetApp storage controller (filer1) through either SAN or NAS technology.

The SAP system ID (SID) of the source system is <<var\_sapinstance\_source\_sid>>, which is also the name of the Sybase ASE database used by this system. Physically, this database resides on two FlexVol<sup>®</sup> volumes on filer1, sapdata\_<<var\_sapinstance\_source\_sid>> and saplog\_<<var\_sapinstance\_source\_sid>>. The Sybase ASE master database also resides on saplog\_<<var\_sapinstance\_source\_sid>>.

The goal of this solution is to clone the volumes containing the data for database <<var\_sapinstance\_source\_sid>> and seamlessly integrate their FlexClones volumes into the cloned system. In order to do this, it's crucial to rename the database to match the SID on the cloned system. In our example, the SID of the cloned system is <<var\_sapinstance\_cloned\_sid>>.

All of the required steps are automated by Snap Creator, which makes the process more efficient and less prone to errors.

### 3.2 Environment

This section describes the environment required to perform a SAP on Sybase database clone and the necessary steps. We assume the architecture consists of two servers, the source and cloned system, with a NetApp storage system attached to both.

#### SAP System Installation

Both the source and the cloned system must be running the same version of the SAP on Sybase system. For installation instructions, refer to the SAP installation guide for your product. For instructions regarding the installation of SAP software with NetApp storage, refer to the installation section in [TR-3442: SAP with Oracle on UNIX and NFS and NetApp Storage](#) or [TR-3533: SAP with Oracle on UNIX and FCP and NetApp Storage](#).

**Note:** The source and the cloned systems' SAP instances must use the same SAP master password.

#### Volume Layout

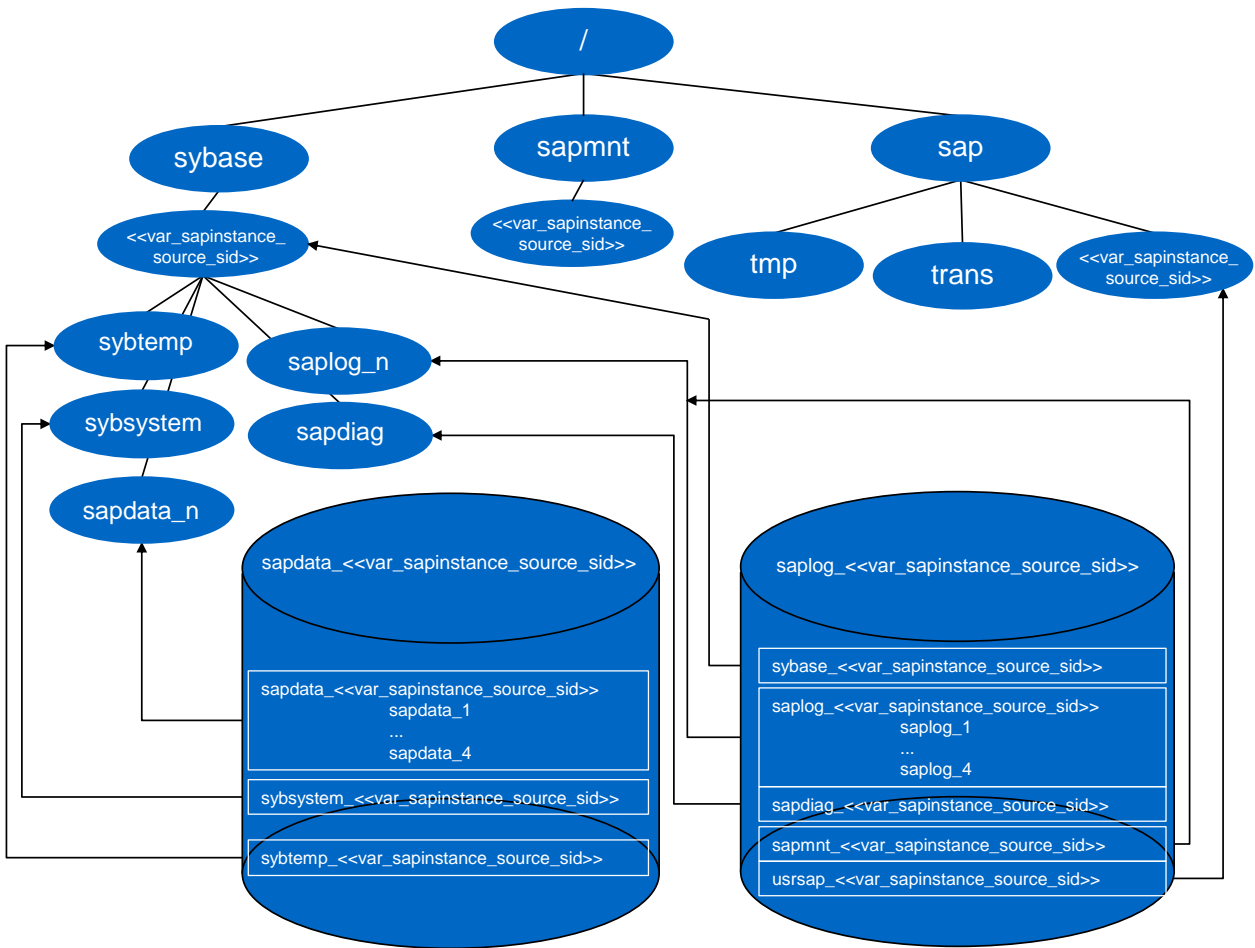
The volume layout is based on the one described in [TR-3442: SAP with Oracle on UNIX and NFS and NetApp Storage](#). Two volumes are created for each the source and cloned system, saplog\_SID, and sapdata\_SID, where SID is the SAP system ID of the system to which the volumes will be exported (for example, <<var\_sapinstance\_source\_sid>> for the source and <<var\_sapinstance\_cloned\_sid>> for the cloned system). Table 1 lists all the qtrees and subdirectories to be created and where to mount them, and Figure 4 illustrates the directories in our example source system, <<var\_sapinstance\_source\_sid>>.

Table 1) FlexVol and qtree layout.

FlexVol Volume	Qtree	Subdirectory to Be Mounted	Mountpoint at SAP Server
saplog_SID	sapmnt_SID		/sapmnt/SID
	usrsap_SID		/usr/sap/SID
	sybase_SID		/sybase/SID
	sapdiag_SID		/sybase/SID/sapdiag
	saplog_SID	saplog_1	/sybase/SID/saplog_1
		saplog_2	/sybase/SID/saplog_2
		saplog_3	/sybase/SID/saplog_3
		saplog_4	/sybase/SID/saplog_4
sapdata_SID	sapdata_SID	sapdata_1	/sybase/SID/sapdata_1
		sapdata_2	/sybase/SID/sapdata_2
		sapdata_3	/sybase/SID/sapdata_3

FlexVol Volume	Qtree	Subdirectory to Be Mounted	Mountpoint at SAP Server
		sapdata_4	/sybase/SID/sapdata_4
	sybsystem_SID		/sybase/SID/sybsystem
	sybtemp_SID		/sybase/SID/sybtemp

Figure 4) SAP directories of the example source system.



## NFS Mount Options

When planning to use NFS, configuring the correct NFS mount options is important to providing optimal performance and system stability. There are common mount options that are valid for all operating system platforms. Some common mount options include `rw`, `bg`, `hard`, `nointr`, `rsize=32768`, `wsizes=32768`, `vers=3`, and `suid`. For additional information, refer to the [NetApp Knowledgebase](#).

Additional mount options for the different operating system platforms include:

- **Linux**<sup>®</sup>. `<common>`, `tcp`, `timeo=600`
- **Solaris**. `<common>`, `proto=tcp`, [`forcdirectio` or `llock`]
- **HP/UX**. `<common>`, `proto=tcp`, `timeo=600`
- **AIX**. `<common>`, `proto=tcp`, `timeo=600`, `cio`

## Snap Creator Installation

For instructions on installing Snap Creator, refer to the [NetApp Snap Creator Framework 3.5.0 Installation and Administration Guide](#).

## Snap Creator Plug-In Installation

To install the Sybase plug-in of Snap Creator that is required to perform backup and clone operations on a Sybase database using Snap Creator, create a folder called `SYBASECLONING` in the plug-ins directory of the Snap Creator server and every Snap Creator agent involved and copy the `SYBASECLONING.pm` file into that folder.

## Creation of Additional Directories

During the cloning process, the qtrees containing the Sybase system and SAP `diag` databases of the source system must be temporarily mounted on the cloned system under the path where they reside on the source system. Therefore, the corresponding folders must be created on the cloned system. In our example, the following folders should be created on the cloned system:

- `/sybase/⟨⟨var_sapinstance_source_sid⟩⟩/sybsystem`
- `/sybase/⟨⟨var_sapinstance_source_sid⟩⟩/sybtemp`
- `/sybase/⟨⟨var_sapinstance_source_sid⟩⟩/sapdiag`

Figure 5 shows how the Sybase directory should look after cloning.

## 4 Snap Creator Configuration

Two types of files are used to configure Snap Creator. The main configuration takes place on the system running the server instance, `scServer` in our example. Several configuration files can exist, but only one of them is used for each operation. Profiles are directories in which one or more configuration files are stored. You can create a new profile by simply creating a new folder under `/path/to/scServer/configs`. NetApp recommends naming the profile and configuration file after the application you want to back up. In this example, `sybasecloning` is the name used for both the profile and the configuration file.

A default configuration template is located in `/path/to/scServer/configs/default/default.conf`. You can copy that template into your profile, rename it, and then modify it to suit your needs.

In addition to the server configuration file, every agent has its own `agent.conf` file. This file is used to tell the agent what operating system commands the server is allowed to execute on that system.

Refer to the Sample Configuration section for an example configuration file based on the architecture described in this report.

### 4.1 Mount and Unmount Commands

In the configuration file, you can specify mount and unmount commands to automatically mount the cloned volumes and their qtrees onto the cloned system. The syntax for mount and unmount commands is `MOUNT_CMDXX=` and `UMOUNT_CMDXX=`, respectively, where `XX` is an ascending number starting from 01. After the equal sign, enter the command your operating system uses to mount the volumes.

**Note:** The commands `mount` and `unmount` require specific OS privileges.

As of Snap Creator 3.5.0, the name of the cloned volume is `cl_⟨profile name⟩_⟨volume name⟩_⟨timestamp of snapshot⟩`. The variable `%SNAP_TIME` can be used to determine the time stamp of the NetApp Snapshot™ copy.



For example, to mount the cloned `sapdata_1` of the source system `<<var_sapinstance_source_sid>>` onto the cloned system `<<var_sapinstance_cloned_sid>>`, the following line should be added to the server configuration file, which, in our example, resides in a profile named `sybasecloning`:

```
MOUNT_CMD01=mount \  
filer1:/vol/cl_sybasecloning_sapdata_<<var_sapinstance_source_sid>>_%SNAP_TIME/sapdata_<<var_sapinstance_source_sid>>/sapdata_1 \  
/sybase/<<var_sapinstance_cloned_sid>>/sapdata_1
```

**Note:** The mount command shown in this example is run in a Linux environment using NFS to mount the cloned volumes. If you are using a different operating system or a different protocol to connect to the storage system, replace the command with the corresponding command for your operating system.

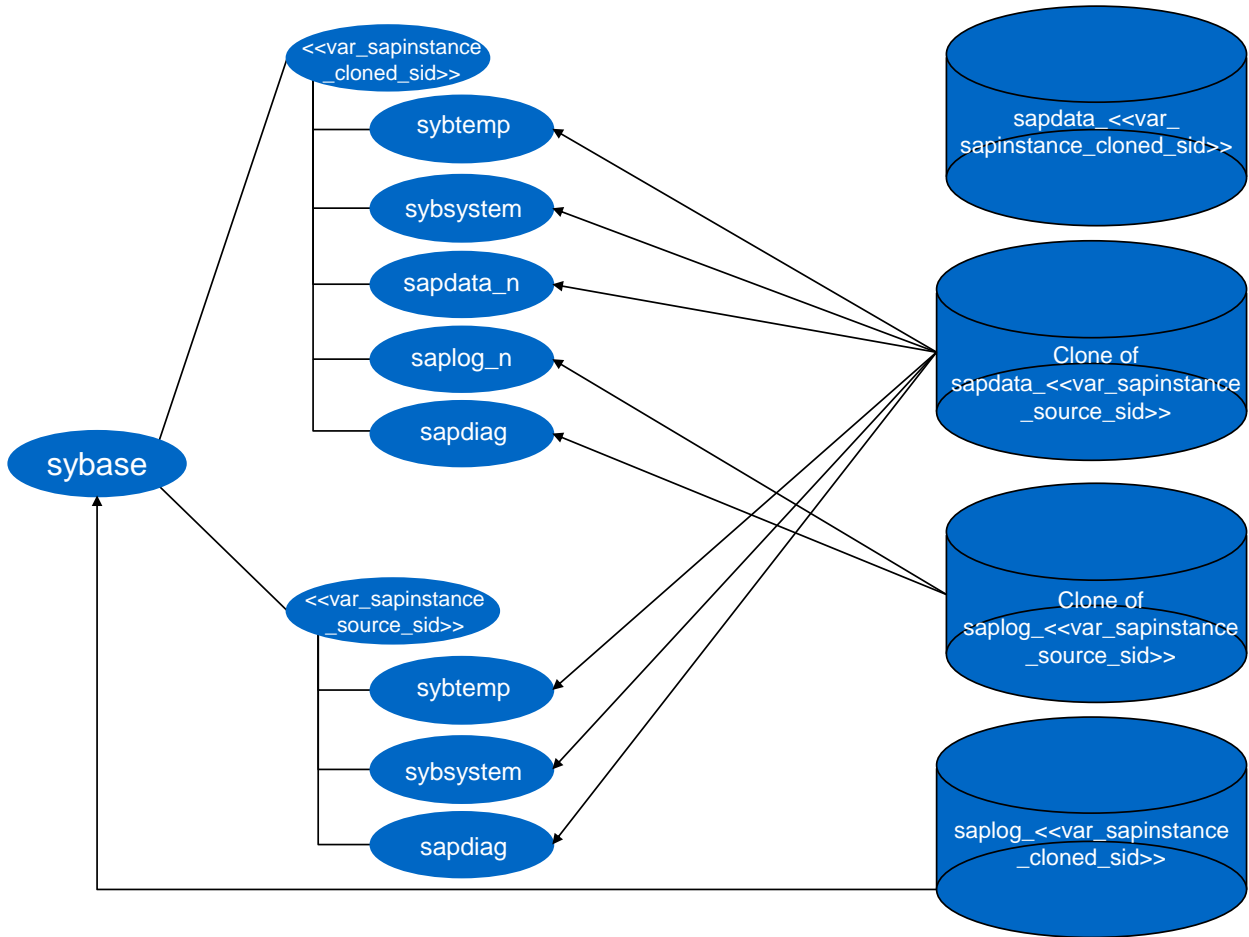
The following qtrees and directories of the cloned volumes must be mounted onto the cloned system, according to Table 1:

- `sapdata_1` through `sapdata_4`
- `saplog_1` through `saplog_4`
- `sybssystem`
- `sybtemp`
- `sapdiag`

Additionally, you must mount the cloned `sybssystem`, `sybtemp`, and `sapdiag` directories under the paths where they reside on the source system (refer to the [Creation of Additional Directories](#) section). This is only necessary for the postclone routine of Snap Creator, and the directories can be unmounted after the cloning is complete. To do this automatically, specify the according unmount commands as `APP_CLONE_FOLLOW_UP_CMDXX=`, where `XX` is an ascending number, starting from 01.

Figure 5 shows which directories should be mounted using the `mount` commands in the server configuration file.

Figure 5) Sybase directory after cloning.



## 4.2 Server Configuration Parameters

The tables in this section list the essential configuration parameters that must be set in the server configuration file in order to perform a clone. For a complete list of parameters, refer to the [NetApp Snap Creator Framework 3.5.0 Installation and Administration Guide](#).

**Note:** All entries in the setting column are example settings that work with the architecture depicted in Figure 3. If there is more than one entry in that column (separated by a slash), then any of the listed options will suffice.

Table 2) Required options.

Variables	Setting	Description
CONFIG_TYPE	PLUGIN/ STANDARD	Use multiple plug-in configurations to build up complex quiesce and unquiesce backup workflows. For a simple clone, operation standard suffices.

Variables	Setting	Description
SNAME	sybasecloningsnap	Snapshot copy name. Must be unique, because Snapshot copies on NetApp storage are deleted according to the naming convention and retention policy used.
SNAP_TIMESTAMP_ONLY	Y/N	If set to Y, Snapshot copies end with YYYYMMDDHHMMSS. Otherwise, the most recent Snapshot copy is renamed to end with recent.
VOLUMES	filer1:sapdata_<<var_sapinstance_source_sid>>,saplog_<<var_sapinstance_source_sid>>	List of source appliances and volumes of which you want to create a Snapshot copy, that is, <Storage system with source database>:<volume1 with source database>, <volume2 with source database>;<filer2>, and so on.
NTAP_SNAPSHOT_RETENTION_S	daily:7	Determines the number of Snapshot copies you want to retain for a given policy. Example setting options include daily:7, weekly:4, and monthly:1.
NTAP_USERS	filer1:exampleuser/<password of exampleuser>	The list of appliances and their corresponding user names and passwords. For example, filer1:joe/password1; filer2:bob/password2; filer3:ken/password3.
NTAP_PWD_PROTECTION	Y/N	Enables password protection. You must encrypt all passwords (storage system, application, and plug-ins) and save encrypted passwords in the configuration file when this option is set to Y. To encrypt passwords, start Snap Creator with the --cryptpasswd flag and the password. If not encrypted, your password will appear in the log files.
TRANSPORT	HTTP/HTTPS	Allows you to choose the protocol for API communications. <b>Note:</b> HTTPS might require openssl-devel libraries.

Variables	Setting	Description
PORT	80	The port you want to use to communicate with the NetApp storage controllers, normally 80/443.
LOG_NUM	30	The number of .debug and .out reports that Snap Creator must retain.

**Table 3) Cloning options.**

Variables	Setting	Description
NTAP_VOL_CLONE_RESERVE	none/file/volume	The space reserved for a cloned volume.
NTAP_CLONE_FOR_BACKUP	N	Setting that determines when the clone is deleted. Must be set to N so the clone will not be deleted until the next run of Snap Creator.

**Note:** The options shown in Table 4 can be omitted, if the cloned volumes are to be mounted in any way that other than through NFS.

**Table 4) NFS-specific options.**

Variables	Setting	Description
NTAP_NFS_EXPORT_HOST	<<var_cloned_system_hostname >>	The hostname or IP address of the machine to which the clone is exported. This is the host where the user wants to mount the cloned volume using NFS.
NTAP_NFS_EXPORT_ACCESS	root	The host specified in NTAP_NFS_EXPORT_HOST receives access or permission to the cloned volume.
NTAP_NFS_EXPORT_PERSISTENT	true	Determines if the NFS export is persistent.

Table 5) Client/server configuration.

Variables	Setting	Description
SC_AGENT	<<var_source_system_hostname>>:9090	Hostname or IP address of the source system and the port on which the Snap Creator agent listens (9090, by default), separated by a colon. Used to perform tasks on remote machines. Must be omitted if Snap Creator server is running on the source system.
SC_CLONE_TARGET	<<var_cloned_system_hostname>>:9090	Hostname or IP address of the cloned system and the port on which the Snap Creator agent is listening (9090, by default), separated by a colon.

Table 6) Sybase settings.

Variables	Setting	Description
APP_NAME	SYBASECLONING	Specifies which application plug-in should be used.
SYBASE_SERVER	<<var_sapinstance_source_sid>>	Name of the Sybase ASE instance running on the source system.
SYBASE_DATABASES	master:sapsa/<sap master password>;<<var_sapinstance_source_sid>>:sapsa/<sap master password>;model:sapsa/<sap master password>;sptools:sapsa/<sap master password>;sybmgmtldb:sapsa/<sap master password>;sybssystemdb:sapsa/<sap master password>;sybssystemprocs:sapsa/<sap master password>	All databases to be cloned, including the SAP and system databases (except tempdb), the username of the owner of the database, and the user's password. The database autodiscovery feature provided by Snap Creator is not supported for cloning. Encrypted passwords are supported if NTAP_PWD_PROTECTION is set. The format is DB1:username/password;DB2:username/password.
SYBASE_ISQL_CMD	/path/to/isql_on_source_system	The path to the isql command on the source system, including the binary file.
SYBASE	/path/to/Sybase/<<var_sapinstance_source_sid>>	The path to the root directory of the Sybase installation on the source system.
SYBASE_TARGET_SSO	sapsso:<sap master password>	Login name and password of the Sybase user owning the system security officer (SSO) role on the cloned system. Format is

Variables	Setting	Description
		login:passwd.
SYBASE_TARGET_SERVER	<<var_sapinstance_cloned_sid>>	Name of the Sybase ASE instance running on the cloned system.
SYBASE_TARGET_RUNSCRIPT	/path/to/RUN_<<var_sapinstance_cloned_sid>>	The path to the RUN_<servername> script used to start Sybase ASE on the cloned system, including the script itself.
SYBASE_TARGET_ISQL_CMD	/path/to/isql_on_target_system	The path to the isql command on the cloned system, including the binary file.
SYBASE_TARGET_LOGIN	sybs20:sapsys	Username and group of the operation system user that installed Sybase. SAP uses username syb<sid> for that user.
SYBASE_TARGET_DATABASE_DEVICEXX	SYBASE_TARGET_DATABASE_DEVICE01=master:master,/sybase/<<var_sapinstance_cloned_sid>>/sybssystem/master.dat; SYBASE_TARGET_DATABASE_DEVICE02=<<var_sapinstance_source_sid>>:data<<var_sapinstance_source_sid>>_1,/sybase/<<var_sapinstance_cloned_sid>>/sapdata_1/data<<var_sapinstance_source_sid>>_1.dat:log<<var_sapinstance_source_sid>>_1,/sybase/<<var_sapinstance_cloned_sid>>/saplog_1/log<<var_sapinstance_source_sid>>_1.dat; SYBASE_TARGET_DATABASE_DEVICE03=model:master,/sybase/<<var_sapinstance_cloned_sid>>/sybssystem/master.dat; SYBASE_TARGET_DATABASE_DEVICE04=saptools:datasaptools_1,/sybase/<<var_sapinstance_cloned_sid>>/sapdiag/datasaptools_1.dat:logsaptools_1,/sybase/<<var_sapinstance_cloned_sid>>/sapdiag/logsaptools_1.dat; SYBASE_TARGET_DATABASE_DEVICE05=sybmgmtdb:sybmgmtdev,/sybase/<<var_sapinstance_cloned_sid>>/sybssystem/sybmgmtdb.dat; SYBASE_TARGET_DATABASE_DEVICE06=sybssystemdb:master,/sybase/<<var_sapinstance_cloned	Map every database to be cloned to every device it uses and the path that device will have on the cloned system. Specify one parameter for every database, where XX must be replaced by an ascending number from 01 through 99. Format is DBNAME:DEVICENAME1, /path/to/DEVICENAME1:DEVICENAME2: /path/to/DEVICENAME2.

Variables	Setting	Description
	d_sid>>/sybssystem/master.dat:sy stemdbdev,/sybase/<<var_sapinst ance_cloned_sid>>/sybssystem/sy bsysdb.dat; SYBASE_TARGET_DATABASE_ DEVICE07=tempdb:master,/syba se/<<var_sapinstance_cloned_sid >>/sybssystem/master.dat:tempdb dev,/sybase/<<var_sapinstance_c loned_sid>>/sybtemp/tempdbdev. dat;sybssystemprocs:sysprocsdev, /sybase/<<var_sapinstance_clone d_sid>>/sybssystem/sysprocs.dat	
SYBASE_TARGET_ADDITIONAL_PATHS	/sybase/<<var_sapinstance_sourc e_sid>>/sybssystem;/sybase/<<var _sapinstance_source_sid>>/sybte mp;/sybase/SY1/sapdiag	Additional paths to be adapted for the SYBASE_TARGET_LOGIN user and group on the cloned system, but not set as database device paths. Used to adapt the paths for the temporarily mounted databases. Format is /path/one;/path/two.
SYBASE_TARGET_DBID		Specify the new database ID of all cloned databases, except the master database. Format is <name of db1 on source system>:<new id1>;<name of db2 on source system>:<new id2>. <b>Note:</b> For SAP database clones, the database ID must stay the same; therefore, this field must be left empty.
SYBASE_TARGET_DATABASE_NAME	<<var_sapinstance_source_sid>> :<<var_sapinstance_cloned_sid> >	The old and new names of every database you want to rename. Format is DB1:NEWNAME1;DB2:NEWNAME2.
SYBASE_STARTUP_ATTEMPTS	Enter a number if the default value does not suit your needs.	The number of Sybase ASE startup attempts made before the operation is aborted. The default is six. <b>Note:</b> On faster systems, a number less than six can increase performance, but slower systems might require more than six attempts.
SYBASE_STARTUP_DELAY	Enter a number if the default value does not suit your needs.	The time in seconds between every startup attempt of Sybase ASE. The default is 10. <b>Note:</b> On faster systems, a

Variables	Setting	Description
		number lower than 10 can increase performance, but slower systems might require more than 10 seconds.

### 4.3 Agent Configuration

The configuration file for each agent resides in `/path/to/scAgent/config/agent.conf`. If your configuration uses `PRE`, `POST`, or `APP` commands other than the built-in application plug-in commands, these commands must be added to the agent configuration file. In this case, the `mount` and `unmount` commands should be added.

For example, to allow the Linux `mount` command, add the following line:

```
Command:mount
```

Additionally, the wildcard (\*) can be used to allow all commands, but for security reasons, NetApp does not recommend using the wildcard.

### 4.4 Running Snap Creator

Because the interactive SQL (ISQL) tool that is used by Snap Creator needs the Sybase environment variables and the locale set to C, source the `SYBASE.x` script located in the root folder of the Sybase installation and change the locale of the system to C. This must be done on the source system only.

To accomplish this on a source system running Linux and using Bash, run the following commands:

```
source /path/to/Sybase/⟨⟨var_sapinstance_source_sid⟩⟩/SYBASE.sh
export LANG="C"
```

Afterward, run the Snap Creator agent on the source and cloned systems. If you run the Snap Creator server on the source system, no agent has to be started on that machine.

The syntax to run the agent is:

```
/path/to/scAgent/snapcreator --start-agent
```

When all agents are running, start the server by running the following command:

```
/path/to/scAgent/snapcreator --profile <profile> --action clone_vol --policy
<policy>
```

Replace `<profile>` with the name of the folder you created in your configuration file (`sybasecloning` is the name used in the example) and `<policy>` with one of the policies you specified in `NTAP_SNAPSHOT_RETENTIONS`. If the name of your configuration file differs from the name of your profile, add this additional flag:

```
--config <config file name>
```

**Note:** For the operation to finish successfully, it is important that both the Sybase database and the SAP instance on the cloned system are shut down beforehand.



## 5 Conclusion

As database sizes and the number of required test and development systems in a SAP landscape grow, the cost to provide all systems with up-to-date data increases heavily. Cloning SAP on Sybase databases with Snap Creator simplifies and accelerates the process using NetApp technologies.

FlexClone provides a method to copy a database in a way that is highly efficient in terms of time and required storage capacity. The Snap Creator Framework triggers application-consistent FlexClone copies and seamlessly integrates them into different systems' SAP installations.

By combining the benefits of FlexClone and the Snap Creator Framework, the steps described in this document provide a method to deploy SAP on a Sybase database to different SAP instances. Using this method saves storage space and time, allows for consistency of the copied databases, and can add significant value to any business running SAP on Sybase software. Once the setup and configuration is completed, all necessary steps are automatically managed by Snap Creator and are therefore less error prone and time consuming.

## Appendix

This section provides additional information about cloning SAP on Sybase databases with Snap Creator.

### Sample Configuration

```
#####
# sybasecloning.conf                                     #
#                                                       #
# Snapcreator example server config file for cloning SAP #
# on Sybase databases, according to TR-XXXX.           #
#####

# Required options

CONFIG_TYPE=STANDARD
SNAME=sybasecloningsnap
SNAP_TIMESTAMP_ONLY=Y
VOLUMES=filer1:sapdata_<<var_sapinstance_source_sid>>,saplog_<<var_sapinstance_source_
sid>>
NTAP_SNAPSHOT_RETENTIONS=daily:7
NTAP_USERS=filer1:exampleuser/<password of exampleuser>
NTAP_PWD_PROTECTION=N
TRANSPORT=HTTP
PORT=80
LOG_NUM=30

# Cloning options

NTAP_VOL_CLONE_RESERVE=none
NTAP_CLONE_FOR_BACKUP=N

# NFS SPECIFIC OPTIONS
NTAP_NFS_EXPORT_HOST=<<var_cloned_system_hostname>>
NTAP_NFS_EXPORT_ACCESS=root
NTAP_NFS_EXPORT_PERSISTENT=true

# Client/Server configuration
SC_AGENT=<<var_source_system_hostname>>:9090
SC_CLONE_TARGET=<<var_cloned_system_hostname>>:9090

# Sybase settings
APP_NAME=SYBASECLONING
```

```

SYBASE_SERVER=<<var_sapinstance_source_sid>>
SYBASE_DATABASES=master:sapsa/<sap master
password>;<<var_sapinstance_source_sid>>:sapsa/<sap master password>;model:sapsa/<sap
master password>;saptools:sapsa/<sap master password>;sybmgmtdb:sapsa/<sap master
password>;sybsystemdb:sapsa/<sap master password>;sybsystemprocs:sapsa/<sap master
password>;tempdb:sapsa/<sap master password>
SYBASE_ISQL_CMD=/sybase/<<var_sapinstance_source_sid>>/OCS-15_0/bin/isql
SYBASE=/sybase
SYBASE_TARGET_SSO=sapssso:<sap master password>
SYBASE_TARGET_SERVER=<<var_sapinstance_cloned_sid>>
SYBASE_TARGET_RUNSCRIPT=/sybase/<<var_sapinstance_cloned_sid>>/ASE-
15_0/install/RUN <<var_sapinstance_cloned_sid>>
SYBASE_TARGET_ISQL_CMD=/sybase/<<var_sapinstance_cloned_sid>>/OCS-15_0/bin/isql
SYBASE_TARGET_LOGIN=sybs12:sapsys
SYBASE_TARGET_DATABASE_DEVICES=master:master,/sybase/<<var_sapinstance_cloned_sid>>/sy
bssystem/master.dat;<<var_sapinstance_source_sid>>:data<<var_sapinstance_source_sid>>_1
,/sybase/<<var_sapinstance_cloned_sid>>/sapdata_1/data<<var_sapinstance_source_sid>>_1
.dat:log<<var_sapinstance_source_sid>>_1,/sybase/<<var_sapinstance_cloned_sid>>/saplog
_1/log<<var_sapinstance_source_sid>>_1.dat:model:master,/sybase/<<var_sapinstance_clon
ed_sid>>/sybsystem/master.dat;saptools:datasaptools_1,/sybase/<<var_sapinstance_cloned
_sid>>/sapdiag/datasaptools_1.dat:logsaptools_1,/sybase/<<var_sapinstance_cloned_sid>>
/sapdiag/logsaptools_1.dat;sybmgmtdb:sybmgmtdev,/sybase/<<var_sapinstance_cloned_sid>>
/sybsystem/sybmgmtdb.dat;sybsystemdb:master,/sybase/<<var_sapinstance_cloned_sid>>/syb
ssystem/master.dat:systemdbdev,/sybase/<<var_sapinstance_cloned_sid>>/sybsystem/sybsysd
b.dat;tempdb:master,/sybase/<<var_sapinstance_cloned_sid>>/sybsystem/master.dat:tempdb
dev,/sybase/<<var_sapinstance_cloned_sid>>/sybtemp/tempdbdev.dat;sybsystemprocs:syspro
csdev,/sybase/<<var_sapinstance_cloned_sid>>/sybsystem/sysprocs.dat
SYBASE_TARGET_ADDITIONAL_PATHS=/sybase/<<var_sapinstance_source_sid>>/sybsystem;/sybas
e/<<var_sapinstance_source_sid>>/sybtemp;/sybase/SY1/sapdiag
SYBASE_TARGET_DBID=
SYBASE_TARGET_DATABASE_NAME=<<var_sapinstance_source_sid>>:<<var_sapinstance_cloned_si
d>>
SYBASE_STARTUP_ATTEMPTS=
SYBASE_STARTUP_DELAY=

# Mount commands

MOUNT_CMD01=mount
filer1:/vol/cl_sybasecloning_sapdata <<var_sapinstance_source_sid>>_%SNAP_TIME/sapdata
<<var_sapinstance_source_sid>>/sapdata_1
/sybase/<<var_sapinstance_cloned_sid>>/sapdata_1
MOUNT_CMD02=mount
filer1:/vol/cl_sybasecloning_sapdata <<var_sapinstance_source_sid>>_%SNAP_TIME/sapdata
<<var_sapinstance_source_sid>>/sapdata_2
/sybase/<<var_sapinstance_cloned_sid>>/sapdata_2
MOUNT_CMD03=mount
filer1:/vol/cl_sybasecloning_sapdata <<var_sapinstance_source_sid>>_%SNAP_TIME/sapdata
<<var_sapinstance_source_sid>>/sapdata_3
/sybase/<<var_sapinstance_cloned_sid>>/sapdata_3
MOUNT_CMD04=mount
filer1:/vol/cl_sybasecloning_sapdata <<var_sapinstance_source_sid>>_%SNAP_TIME/sapdata
<<var_sapinstance_source_sid>>/sapdata_4
/sybase/<<var_sapinstance_cloned_sid>>/sapdata_4

MOUNT_CMD05=mount
filer1:/vol/cl_sybasecloning_saplog <<var_sapinstance_source_sid>>_%SNAP_TIME/saplog <
<var_sapinstance_source_sid>>/saplog_1 /sybase/<<var_sapinstance_cloned_sid>>/saplog_1
MOUNT_CMD06=mount
filer1:/vol/cl_sybasecloning_saplog <<var_sapinstance_source_sid>>_%SNAP_TIME/saplog <
<var_sapinstance_source_sid>>/saplog_2 /sybase/<<var_sapinstance_cloned_sid>>/saplog_2
MOUNT_CMD07=mount
filer1:/vol/cl_sybasecloning_saplog <<var_sapinstance_source_sid>>_%SNAP_TIME/saplog <
<var_sapinstance_source_sid>>/saplog_3 /sybase/<<var_sapinstance_cloned_sid>>/saplog_3

```

```

MOUNT_CMD08=mount
filer1:/vol/cl_sybasecloning_saplog_<<var_sapinstance_source_sid>>_%SNAP_TIME/saplog_<
<var_sapinstance_source_sid>>/saplog_4 /sybase/<<var_sapinstance_cloned_sid>>/saplog_4

MOUNT_CMD09=mount
filer1:/vol/cl_sybasecloning_sapdata_<<var_sapinstance_source_sid>>_%SNAP_TIME/sybsyst
em_<<var_sapinstance_source_sid>> /sybase/<<var_sapinstance_cloned_sid>>/sybsystem
MOUNT_CMD10=mount
filer1:/vol/cl_sybasecloning_sapdata_<<var_sapinstance_source_sid>>_%SNAP_TIME/sybtemp
_<<var_sapinstance_source_sid>> /sybase/<<var_sapinstance_cloned_sid>>/sybtemp

MOUNT_CMD11=mount
filer1:/vol/cl_sybasecloning_saplog_<<var_sapinstance_source_sid>>_%SNAP_TIME/sapdiag_
<<var_sapinstance_source_sid>> /sybase/<<var_sapinstance_cloned_sid>>/sapdiag

# Mounting the temporary directories

MOUNT_CMD12=mount
filer1:/vol/cl_sybasecloning_sapdata_<<var_sapinstance_source_sid>>_%SNAP_TIME/sybsyst
em_<<var_sapinstance_source_sid>> /sybase/<<var_sapinstance_source_sid>>/sybsystem
MOUNT_CMD13=mount
filer1:/vol/cl_sybasecloning_sapdata_<<var_sapinstance_source_sid>>_%SNAP_TIME/sybtemp
_<<var_sapinstance_source_sid>> /sybase/<<var_sapinstance_source_sid>>/sybtemp
MOUNT_CMD14=mount
filer1:/vol/cl_sybasecloning_saplog_<<var_sapinstance_source_sid>>_%SNAP_TIME/sapdiag_
<<var_sapinstance_source_sid>> /sybase/<<var_sapinstance_source_sid>>/sapdiag

# Unmount commands, followed by ";echo "umount"" so they won't fail if the
# directories are not mounted

UMOUNT_CMD01=umount -l /sybase/<<var_sapinstance_cloned_sid>>/sapdata_1;echo "umount"
UMOUNT_CMD02=umount -l /sybase/<<var_sapinstance_cloned_sid>>/sapdata_2;echo "umount"
UMOUNT_CMD03=umount -l /sybase/<<var_sapinstance_cloned_sid>>/sapdata_3;echo "umount"
UMOUNT_CMD04=umount -l /sybase/<<var_sapinstance_cloned_sid>>/sapdata_4;echo "umount"

UMOUNT_CMD05=umount -l /sybase/<<var_sapinstance_cloned_sid>>/saplog_1;echo "umount"
UMOUNT_CMD06=umount -l /sybase/<<var_sapinstance_cloned_sid>>/saplog_2;echo "umount"
UMOUNT_CMD07=umount -l /sybase/<<var_sapinstance_cloned_sid>>/saplog_3;echo "umount"
UMOUNT_CMD08=umount -l /sybase/<<var_sapinstance_cloned_sid>>/saplog_4;echo "umount"

UMOUNT_CMD09=umount -l /sybase/<<var_sapinstance_cloned_sid>>/sybsystem;echo "umount"
UMOUNT_CMD10=umount -l /sybase/<<var_sapinstance_cloned_sid>>/sybtemp;echo "umount"

UMOUNT_CMD11=umount -l /sybase/<<var_sapinstance_cloned_sid>>/sapdiag;echo "umount"

# Unmount temporary directories before cloning, in case something went wrong in the
last run of snapcreator

UMOUNT_CMD12=umount -l /sybase/<<var_sapinstance_source_sid>>/sybsystem;echo "umount"
UMOUNT_CMD13=umount -l /sybase/<<var_sapinstance_source_sid>>/sybtemp;echo "umount"
UMOUNT_CMD14=umount -l /sybase/<<var_sapinstance_source_sid>>/sapdiag;echo "umount"

# Unmount the tempory directories after sucessfull cloning, using
APP_CLONE_FOLLOW_UP_CMDS

APP_CLONE_FOLLOW_UP_CMD01=umount -l
/sybase/<<var_sapinstance_source_sid>>/sybsystem;echo "follow up"
APP_CLONE_FOLLOW_UP_CMD02=umount -l
/sybase/<<var_sapinstance_source_sid>>/sybtemp;echo "follow up"
APP_CLONE_FOLLOW_UP_CMD03=umount -l
/sybase/<<var_sapinstance_source_sid>>/sapdiag;echo "follow up"

```

## Variables Used in this Document

Table 7) Variable definitions.

Variable Name	Meaning
<<var_sapinstance_source_sid>>	SAP identifier of the SAP instance to be cloned.
<<var_sapinstance_cloned_sid>>	SAP identifier of the SAP instance to receive the cloned database.
<<var_source_system_hostname>>	Host name of the system running the SAP instance to be cloned.
<<var_cloned_system_hostname>>	Host name of the system running the SAP instance to receive the cloned database.

## References

This report builds on the work of others and includes concepts that can be explored in more detail. The following documents either were used as references in this report or can provide additional details on the topics discussed in this report.

- [RA-0005: Accelerating Development of Microsoft SQL Applications in Heterogeneous Environments](#)
- [TR-3442: SAP with Oracle on UNIX and NFS and NetApp Storage](#)
- [TR-3533: SAP with Oracle on UNIX and FCP and NetApp Storage](#)
- [NetApp Snap Creator Framework 3.5.0 Installation and Administration Guide](#)

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®



[www.netapp.com](http://www.netapp.com)

© 2012 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexClone, FlexVol, Snap Creator, and Snapshot are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. SAP is a registered trademark of SAP AG. Microsoft and SQL Server are registered trademarks of Microsoft Corporation. Oracle is a registered trademark of Oracle Corporation. UNIX is a registered trademark of The Open Group. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-4038-0312